

# Components

## Schemas

### AnimalSpeciesDTO

DTO representing an animal species

*Properties*

Name	Description	Schema
speciesName <i>required</i>	Species name	string

### AnimalDTO

DTO representing an animal

*Properties*

Name	Description	Schema
name <i>required</i>	Name of the animal	string
age <i>required</i>	Age of the animal	integer (int32)
animalSpecies <i>required</i>	Species of the animal	string
description <i>required</i>	Description of the animal	string
needsAttention <i>required</i>		boolean
needsOutdoorSpace <i>required</i>		boolean
goodWithChildren <i>optional</i>		boolean
socialAnimal <i>optional</i>		boolean

### AdoptionStatusDTO

*Properties*

Name	Description	Schema
adopterName <i>required</i>	Adopter's name	string

Name	Description	Schema
animalName <i>required</i>	Animal's name	string
status <i>required</i>		string

## PutAdopterDto

DTO for putting adopter

*Properties*

Name	Description	Schema
phone <i>required</i>	Adopter's phone number	string
street <i>required</i>	Adopter's street	string
city <i>required</i>	Adopter's city	string
state <i>required</i>	Adopter's state	string
zip <i>required</i>	Adopter's zip	string
country <i>required</i>	Adopter's country	string

## RoleRequestDTO

Details to change role

*Properties*

Name	Description	Schema
user <i>optional</i>	User email	string
role <i>optional</i>	Role name	string
operation <i>optional</i>	Role operation	enum (GRANT,REMOVE)

## UserRegisterDTO

Details to create user

*Properties*

Name	Description	Schema
name <i>optional</i>	User's name	string
lastname <i>optional</i>	User's lastname	string
email <i>optional</i>	User's email	string

Name	Description	Schema
password <i>optional</i>	User's password	string

## AuthenticationDTO

Details to authenticate

*Properties*

Name	Description	Schema
email <i>optional</i>	User email	string
password <i>optional</i>	User password	string

## AuthenticationResponse

*Properties*

Name	Description	Schema
jwt <i>optional</i>		string

## ChangePassDTO

New password

*Properties*

Name	Description	Schema
new_password <i>optional</i>	New Password	string

## PasswordChanged

*Properties*

Name	Description	Schema
email <i>optional</i>		string
status <i>optional</i>		string

## AdoptionFormDTO

Adoption Form Data Transfer Object

*Properties*

Name	Description	Schema
hasBigGarden <i>optional</i>	is the adopter's house big enough for the animal?	boolean
hasOtherPets <i>optional</i>	is the adopter have other pets?	boolean
hasMuchFreeTime <i>optional</i>	is the adopter have much free time?	boolean
hasChildren <i>optional</i>	is the adopter have children?	boolean
preferredAge <i>optional</i>	animal age for adoption	integer (int32)
preferredSpecies <i>optional</i>	animal name for adoption	string

## CreateAdopterDTO

DTO representing a new adopter

*Properties*

Name	Description	Schema
phone <i>required</i>	Adopter phone	string
street <i>required</i>	Adopter street	string
city <i>required</i>	Adopter city	string
state <i>required</i>	Adopter state	string
zip <i>required</i>	Adopter zip	string
country <i>required</i>	Adopter country	string

## GetAdopterDTO

DTO for getting adopter

*Properties*

Name	Description	Schema
id <i>optional</i>	Adopter ID	integer (int64)
email <i>optional</i>		string

## AnimalSpeciesWithIdDTO

DTO representing an animal species with id

*Properties*

Name	Description	Schema
speciesName <i>required</i>	Species name	string
id <i>optional</i>	Species id	integer (int64)

## AdoptionDTO

Adoption Data Transfer Object

*Properties*

Name	Description	Schema
adopterName <i>required</i>	Adopter's name	string
animalName <i>required</i>	Animal's name	string

## PaginatedResponse

DTO representing a paginated response

*Properties*

Name	Description	Schema
content <i>optional</i>	Content of the page	< object > array
currentPage <i>optional</i>	Total number of pages	integer (int32)
totalElements <i>optional</i>	Total number of elements	integer (int64)
size <i>optional</i>	Total number of pages	integer (int32)

## GetAdopterByIdDTO

DTO for getting adopter by ID

*Properties*

Name	Description	Schema
id <i>optional</i>	Adopter ID	integer (int64)
name <i>optional</i>	Adopter name	string
lastname <i>optional</i>	Adopter lastname	string
email <i>optional</i>	Adopter email	string
phone <i>optional</i>	Adopter phone	string

# OpenAPI definition

vv0 :revnumber: v0 :openapi: 3.0.1

## Overview

## Tags

### Users

Endpoints for user management

### Admin

Endpoints for admin management

### Adoptions

Endpoints for managing adoptions, roles: ADMIN, USER, ADOPTER

### Animal Species

Endpoints for managing animal species

### Adopters

Endpoints for managing adopters

### Animals

Endpoints for managing animals

## Paths

### ***GET* /api/animalspecies/{id}** Find animal species by ID

Returns an AnimalSpeciesDTO object

#### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

#### Responses

Code	Description	Links
409	Species name already exists	No Links
404	Animal species not found	No Links

Code	Description	Links
200	Successfully retrieved the animal species  <i>Content</i> <b>application/json</b>	No Links

## ***PUT* /api/animalspecies/{id} Update animal species**

Updates an existing AnimalSpeciesDTO object

### Parameters

Type	Name	Description	Schema
<b>path</b>	<b>id</b> <i>required</i>		integer (int64)

### Responses

Code	Description	Links
409	Species name already exists	No Links
200	Successfully updated the animal species  <i>Content</i> <b>application/json</b>	No Links
404	Animal species not found	No Links

## ***DELETE* /api/animalspecies/{id} Delete animal species**

Deletes an AnimalSpeciesDTO object by ID

### Parameters

Type	Name	Description	Schema
<b>path</b>	<b>id</b> <i>required</i>		integer (int64)

### Responses

Code	Description	Links
409	Species name already exists	No Links
404	Animal species not found	No Links
204	Successfully deleted the animal species	No Links

## **GET** */api/animal/{id}* Find animal by ID

Returns an AnimalDTO object

### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

### Responses

Code	Description	Links
200	Successfully retrieved the animal  <i>Content</i> <b>application/json</b>	No Links
404	Animal not found	No Links

## **PUT** */api/animal/{id}* Update animal

Updates an existing AnimalDTO object

### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

### Responses

Code	Description	Links
200	Successfully updated the animal  <i>Content</i> <b>application/json</b>	No Links
404	Animal not found	No Links

## **DELETE** */api/animal/{id}* Delete animal

Deletes an AnimalDTO object by ID

### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)



### Responses

Code	Description	Links
204	Successfully deleted the animal	No Links
404	Animal not found	No Links

## **PUT /api/adoptions/adopt/{id}** Update adoption status by id, roles: ADMIN, USER

Returns updated adoption status.

### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

### Responses

Code	Description	Links
500	Internal server error	No Links
200	Successfully update status for adoption  <i>Content</i> application/json	No Links

## **GET /api/adopters** Find all adopters

### Parameters

Type	Name	Description	Schema
query	filter <i>optional</i>		string
query	size <i>optional</i>		integer (int32)
query	page <i>optional</i>		integer (int32)
query	sort <i>optional</i>		string
query	direction <i>optional</i>		string

### Responses

Code	Description	Links
200	Successfully retrieved list of adopters  <i>Content</i> <b>application/json</b>	No Links
500	Internal server error	No Links

## ***PUT* /api/adopters** Update adopter

### Responses

Code	Description	Links
404	Adopter not found	No Links
500	Internal server error	No Links
200	Successfully updated adopter  <i>Content</i> <b>application/json</b>	No Links

## ***POST* /api/adopters** Create adopter

### Responses

Code	Description	Links
500	Internal server error	No Links
201	Successfully created adopter  <i>Content</i> <b>application/json</b>	No Links
400	Bad request	No Links

## ***PUT* /api/admin/user/role** Endpoint for changing user role

### Responses

Code	Description	Links
200	User role changed  <i>Content</i> /	No Links

Code	Description	Links
400	User role not changed  <i>Content</i> /	No Links

## ***POST* /api/auth/signup Register a new user**

### *Responses*

Code	Description	Links
400	User account not created  <i>Content</i> /	No Links
200	User created  <i>Content</i> /	No Links

## ***POST* /api/auth/signin Endpoint for authentication**

### *Responses*

Code	Description	Links
400	User not authenticated  <i>Content</i> /	No Links
200	User authenticated  <i>Content</i> /	No Links

## ***POST* /api/auth/changepass Endpoint for changing password**

### *Responses*

Code	Description	Links
400	Password not changed  <i>Content</i> /	No Links

Code	Description	Links
200	Password changed  <i>Content</i> /	No Links

## **GET /api/animalspecies** Find all animal species

Returns a list of AnimalSpeciesWithIdDTO objects

### Parameters

Type	Name	Description	Schema
query	<b>filter</b> <i>optional</i>		string
query	<b>size</b> <i>optional</i>	Number of elements per page	integer (int32)
query	<b>page</b> <i>optional</i>	Page number, starting from 0	integer (int32)
query	<b>sort</b> <i>optional</i>	Sorting criteria in the format: property(,asc	desc). Default sort order is ascending. Multiple sort criteria are supported.
string	<b>query</b>	<b>direction</b> <i>optional</i>	Sorting order in the format: asc

### Responses

Code	Description	Links
409	Species name already exists	No Links
200	Successfully retrieved list of animal species  <i>Content</i> <b>application/json</b>	No Links
500	Internal server error	No Links

## **POST /api/animalspecies** Create a new animal species

Saves a new AnimalSpeciesDTO object

### Responses

Code	Description	Links
409	Species name already exists	No Links

Code	Description	Links
400	Invalid input	No Links
200	Successfully created the animal species  <i>Content</i> <b>application/json</b>	No Links

## **GET** /api/animal Find all animals

Returns a list of AnimalsWithIdDTO objects

### Parameters

Type	Name	Description	Schema
<b>query</b>	<b>filter</b> <i>optional</i>	Filter criteria in the format: property[.operation]:value. Example: name.eq=Zoe	string
<b>query</b>	<b>size</b> <i>optional</i>	Number of elements per page	integer (int32)
<b>query</b>	<b>page</b> <i>optional</i>	Page number, starting from 0	integer (int32)
<b>query</b>	<b>sort</b> <i>optional</i>	Sorting criteria in the format: property(,asc	desc). Default sort order is ascending. Multiple sort criteria are supported.
string	<b>query</b>	<b>direction</b> <i>optional</i>	Sorting order in the format: asc

### Responses

Code	Description	Links
500	Internal server error	No Links
200	Successfully retrieved list of animals  <i>Content</i> <b>application/json</b>	No Links

## **POST** /api/animal Create a new animal

Saves a new AnimalDTO object

### Responses

Code	Description	Links
400	Invalid input	No Links

Code	Description	Links
200	Successfully created the animal  <i>Content</i> <b>application/json</b>	No Links

## **POST** **/api/adoptions/matchAnimal** Match adopter to animal

Returns a map where key is a string representation of AnimalDTO and value is the match rating as double.

### Parameters

Type	Name	Description	Schema
query	size <i>optional</i>		integer (int32)
query	page <i>optional</i>		integer (int32)

### Responses

Code	Description	Links
500	Internal server error	No Links
200	Successfully matched adopter to animal  <i>Content</i> <b>application/json</b>	No Links

## **POST** **/api/adoptions/adopt/{animalId}** Adopt an animal, roles: ADOPTER

### Responses

Code	Description	Links
500	Internal server error	No Links
201	Successfully adopted animal  <i>Content</i> <b>application/json</b>	No Links
400	Animal not found or Animal already adopted or User not found or User is not an adopter.	No Links

## **GET /api/adoptions** Get all adoptions, roles: ADMIN, USER

### Responses

Code	Description	Links
200	Found the adoptions  <i>Content</i> <b>application/json</b>	No Links

## **GET /api/adoptions/{id}** Get adoption by id, roles: ADMIN, USER

### Parameters

Type	Name	Description	Schema
path	<b>id</b> <i>required</i>		integer (int64)

### Responses

Code	Description	Links
200	Found the adoption  <i>Content</i> <b>application/json</b>	No Links
404	Adoption not found	No Links

## **DELETE /api/adoptions/{id}** Delete adoption by id, roles: ADMIN, USER

### Parameters

Type	Name	Description	Schema
path	<b>id</b> <i>required</i>		integer (int64)

### Responses

Code	Description	Links
204	Successfully deleted adoption  <i>Content</i> <b>application/json</b>	No Links

Code	Description	Links
404	Adoption not found	No Links

## **GET /api/adoptions/myadoptions** Get all adoptions of current user, roles: ADOPTER

### Parameters

Type	Name	Description	Schema
query	size <i>required</i>	Page size	integer (int32)
query	currentPage <i>required</i>	Current page index	integer (int32)

### Responses

Code	Description	Links
200	Found the adoptions  <i>Content</i> <b>application/json</b>	No Links

## **GET /api/adopters/{id}** Find adopter by ID

### Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

### Responses

Code	Description	Links
200	Successfully retrieved adopter  <i>Content</i> <b>application/json</b>	No Links
404	Adopter not found	No Links
500	Internal server error	No Links

## **GET /api/admin/user** Endpoint for getting all users from db

### Responses



Code	Description	Links
400	Users not retrieved  <i>Content</i> /	No Links
200	Users retrieved  <i>Content</i> /	No Links

## ***GET* /api/admin/logs** Endpoint for retrieving all saved logs from db

### Responses

Code	Description	Links
200	Logs retrieved  <i>Content</i> /	No Links
400	Logs not retrieved  <i>Content</i> /	No Links

## ***DELETE* /api/admin/user/{email}** Endpoint for deleting user from db

### Parameters

Type	Name	Description	Schema
path	<b>email</b> <i>required</i>	User email	string

### Responses

Code	Description	Links
200	User deleted  <i>Content</i> /	No Links

Code	Description	Links
400	User not deleted  <i>Content</i> <i>/</i>	No Links